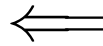


Vypni zvonenie na mobilnom telefóne!

Táto písomná skúška trvá 90 minút. Počas skúšky je povolené používať knihy, poznámky, učebné texty, referencie jazyka C/C++ v papierovej forme. Fotenie tohto dokumentu počas skúšky je zakázané. Kódy predpokladajú kompilátor s podporou C++ 17. Len jedna odpoveď A, B, C, D je správna.

Meno a priezvisko:



1. Koľkokrát bude zavolaný deštruktor triedy Token? (3 b.) Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class Token {
4 public:
5     ~Token() { std::cout << "~Token()"; }
6 };
7
8 void foo(Token t) {}
9
10 int main() {
11     Token t;
12     foo(t);
13
14     return 0;
15 }
```

Riešenie:

- A 0
 - B 1
 - C 2
 - D 3
-

2. Koľkokrát bude zavolaný deštruktor triedy Token? (3 b.) Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class Token {
4 public:
5     ~Token() { std::cout << "~Token()"; }
6 };
7
8 void foo(const Token& t) {}
9
10 int main() {
11     Token t;
12     foo(t);
13
14     return 0;
15 }
```

Riešenie:

- A 0
 - B 1
 - C 2
 - D 3
-

3. Čo sa vypíše na obrazovku? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class Token {
4 public:
5     int a{1};
6     Token(int a0) : a(a0) { std::cout << "Token(int)"; }
7     Token(const Token& t0) { std::cout << "Token(const_Token&)"; }
8 };
9
10 int main() {
11     Token t0(3), t1(4);
12
13     auto greater = [] (Token t0, Token t1) -> bool {
14         return t0.a > t1.a;
15     };
16
17     std::cout << greater(t0, t1);
18
19     return 0;
20 }
```

Riešenie:

- A Token(int)Token(int)0
 - B Token(int)Token(const Token&)Token(int)Token(const Token&)0
 - C Token(int)Token(const Token&)0
 - D Token(int)Token(int)Token(const Token&)Token(const Token&)0
-

4. Čo sa vypíše na obrazovku? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class Token {
4 public:
5     int a{1};
6     Token(int a0) : a(a0) { std::cout << "Token(int)"; }
7     Token(const Token& t0) { std::cout << "Token(const_Token&)"; }
8 };
9
10 int main() {
11     Token t0(3), t1(4);
12
13     auto greater = [&] () -> bool {
14         return t0.a > t1.a;
15     };
16
17     std::cout << greater();
18
19     return 0;
20 }
```

Riešenie:

- A Token(int)Token(int)0
 - B Token(int)Token(const Token&)Token(int)Token(const Token&)0
 - C Token(int)Token(const Token&)0
 - D Token(int)Token(int)Token(const Token&)Token(const Token&)0
-

5. Čo sa vypíše na obrazovku? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class Token {
4 public:
5     int a{1};
6     Token(int a0) : a(a0) { std::cout << "Token(int)"; }
7     Token(const Token& t0) { std::cout << "Token(const_Token&)"; }
8 };
9
10 int main() {
11     Token t0(3), t1(4);
12
13     auto greater = [=] () -> bool {
14         return t0.a > t1.a;
15     };
16
17     std::cout << greater();
18
19     return 0;
20 }
```

Riešenie:

- A Token(int)Token(int)0
 - B Token(int)Token(const Token&)Token(int)Token(const Token&)0
 - C Token(int)Token(const Token&)0
 - D Token(int)Token(int)Token(const Token&)Token(const Token&)0
-

6. Čo sa vypíše na obrazovku? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class Base {
4 public:
5     virtual ~Base() { std::cout << "~Base()"; }
6 };
7
8 class Derived : public Base {
9 public:
10    ~Derived() { std::cout << "~Derived()"; }
11 };
12
13 int main() {
14     Derived d;
15
16     return 0;
17 }
```

Riešenie:

- A ~Base()~Derived()
 - B ~Derived()~Base()
 - C ~Base()
 - D ~Derived()
-

7. Koľkokrát bude zavolaný kopirovací konštruktor triedy Token? Skompiluje? (8 b.) Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 #include <vector>
3
4 class Token {
5 public:
6     Token(const Token& t) { std::cout << "Token(const_Token&); }
7 };
8
9 int main() {
10    std::vector<Token*> v;
11    Token* t = new Token;
12
13    v.push_back(t);
14    v.push_back(t);
15
16    return 0;
17 }
```

Riešenie:

- A 0
 - B 1
 - C 2
 - D neskompiluje
-

8. Nastane alebo nenastane výnimka? (8 bodov)

Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2 #include <map>
3
4 void foo(std::map<int, int>& m, const int a) noexcept {
5     ++m.at(a);
6 }
7
8 int main() {
9     std::map<int, int> m;
10
11     try {
12         foo(m, 10);
13     } catch(...) {
14         std::cout << "Exception";
15     }
16
17     return 0;
18 }
```

Riešenie (len 3 možnosti):

- A nastane výnimka a zachytíme ju pomocou try/catch
 - B nastane výnimka a nevieme ju zachytiť
 - C výnimka nenastane
-

9. Čo sa vypíše na obrazovku? Nastane chyba? (8 bodov) Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class Token {
4 public:
5     int id{0};
6 };
7
8 Token* newToken() {
9     Token t;
10    static int id = 1;
11    t.id = id;
12    ++id;
13
14    return &t;
15 }
16
17 int main() {
18     Token* t = newToken();
19
20     std::cout << t->id;
21
22     return 0;
23 }
```

Riešenie:

- A 0
 - B 1
 - C nedefinované správanie
 - D neskompiluje
-

10. Čo sa vypíše na obrazovku? Nastane chyba? (8 bodov) Písomné odôvodnenie riešenia (min. 5 slov, max. 20 slov):

```
1 #include <iostream>
2
3 class Token0 {
4 friend class Token1;
5
6 private:
7     int a{0};
8 };
9
10 class Token1 {
11 public:
12     int foo(Token0& t0) {
13         t0.a = 1;
14
15         return t0.a;
16     }
17 };
18
19 int main() {
20     Token0 t0;
21     Token1 t1;
22
23     std::cout << t1.foo(t0);
24
25     return 0;
26 }
```

Riešenie:

- A 0
 - B 1
 - C nedefinované správanie
 - D neskompiluje
-